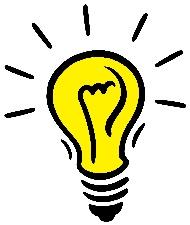
Standards Java

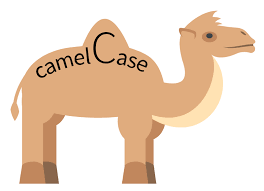
# CamelCase - Simple English Wikipedia, the free encyclopediaVocabulaire



La saviez vous?

Cette photo est erronée. C’est un dromadaire, pas un chameau! Le vrai nom est PascalCase, UpperCamelCase ou DromedaryCase

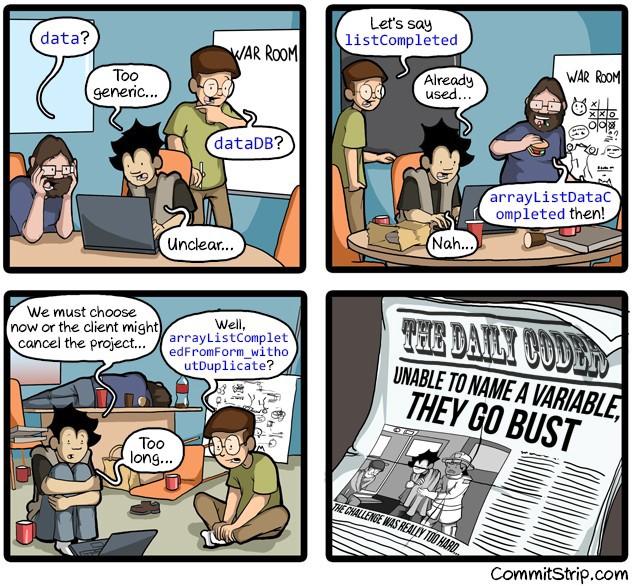
Exemple : PowerPoint, MasterCard



Exemple : iPhone, eBay, johnSmith

# 

# Nommage

* Évite les acronymes/abréviations
* Tu peux écrire les noms en français, mais évite les caractères spéciaux (é, è, ç, â, à, î, …)
* Indentation : 2 à 4 espaces

🡺 IMPORTANT : Rester consistant

* Longueur ligne : 70 à 120 caractères

maximum selon la facilité de lecture

Règles spécifiques

### Packages

Le nom est en **minuscule**

* com.outilsdedefense ✅
* com.outils\_de\_defense ❎
* com.outilsDeDefense ❎

### 

### Méthodes

Le nom est en **camelCase**

Le nom des méthodes doit être un verbe

* calculerTaxe() ✅
* avancer() ✅
* ajouterMontant() ✅
* moyenne() ❎
* GetAge() ❎

## Exemple

package com.example.nomdupackage;

/\*\*  
 \* Document standard, exemple 1  
 \*   
 \* [@author](http://twitter.com/author) Ti-Jo Sanchez   
 \* [@date](http://twitter.com/date) 2020/09/01  
 \* [@version](http://twitter.com/version) 1  
 \* [@copyright](http://twitter.com/copyright) MIT  
 \*   
 \*/

### Classes

Le nom est en **PascalCase**

Le fichier match le nom de la classe publique dedans

Le fichier se termine par .java

* Animal.java ✅
* PersonnaliteConnue.java ✅
* itemDefensif.java ❎

import com.example.util.FileUtil;

import java.util.ArrayList;

public class Personne {  
 public static final Integer NOMBRE\_PARENTS = 2;   
 static final Integer NOMBRE\_COUSINS = 5;   
 private static final Integer NOMBRE\_ANIMAUX = 3;

public String nom;   
 String code\_postal;   
 private String addresse;

public Personne() {}

public Personne(String nom) {  
 this. nom = nom;  
 }

public String obtenirRestoPrefere() {  
 return "Casa Grecque";   
 }

public getAdresse(){

return addresse;

}  
}

### Variables

Le nom est en **minuscule**, en **camelCase** ou **séparé par des \_**

Le nom décrit ce que la variable contient

Le nom ne doit pas être trop détaillé

int schoolId; ✅

int schoolIdentificationNumber; ❎ Trop long

int[] uniqueSchooldIds; ✅

int[] schoolIdsAfterRemovingDuplicates; ❎ Trop long

Map<Integer, User> usersById; ✅

Map<Integer, User> idToUserMap; ❎ Trop d’information inutile

String value; ✅  
String valueString; ❎ Répétition du type

int compteur = 10; ✅

int x = 10; ❎ Trop vague

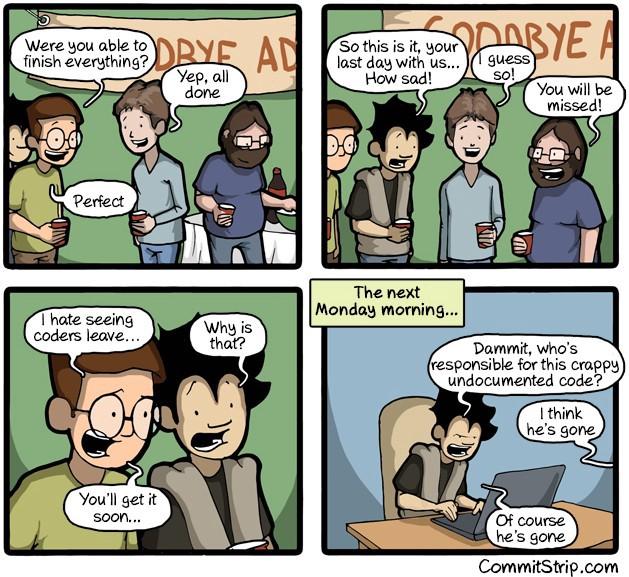
### Constantes

Le nom est en **MAJUSCULE**

Les mots sont séparés avec des \_

* LARGEUR\_MINIMUM ✅
* TAILLE ✅
* Longueur\_totale ❎
* profondeurMax ❎

# Pourquoi?



Tu vas presque toujours rejoindre une équipe qui travaille sur un logiciel déjà existant, dont l’auteur est parti depuis longtemps. Coder en utilisant des standards permet de réduire la frustration de devoir comprendre la structure, en plus de devoir comprendre le code.

Utilise ce document comme une référence, chaque fois que tu te questionnes sur un standard.

## Syntaxe générale

package

[Commentaire] aperçu / résumé de la classe

imports

[Commentaire] information spécifique de la classe (optionnel)

classe

variables statiques en ordre de visibilité

variables non-statiques en ordre de visibilité

// 1. PUBLIQUE

// 2. PROTECTED

// 3. PRIVATE

constructeur vide

constructeur en ordre de paramètres

méthodes

getters, setters, equals, hashCode et toString

